异步图书
www.epubit.com.cn

DATA
MODELING MADE SIMPLE
2ND EDITION

数据建模经典教程（第2版）

[美] Steve Hoberman 著 丁永军 译

商业及IT专业人员的实用指南

中国工信出版集团
人民邮电出版社
POSTS & TELECOM PRESS

目　录

名词解释

欢迎来到异步社区！

版权信息

书名：数据建模经典教程（第2版）

ISBN：978-7-115-45581-9

- 著　　[美] Steve Hoberman

  译　　丁永军

  责任编辑　胡俊英

- 人民邮电出版社出版发行　　北京市丰台区成寿寺路11号

  邮编　100164　电子邮件　315@ptpress.com.cn

网址　http://www.ptpress.com.cn

- 读者服务热线：(010)81055410

反盗版热线：(010)81055315

版权声明

内容提要

数据建模指的是对现实世界各类数据的抽象组织，确定数据库需管辖的范围、数据的组织形式等直至转化成现实的数据库。而数据模型是构建应用系统的核心，是尽可能精准地表示业务运转的概念性框架。

本书通过平实的语言，对数据模型及建模过程进行了深入浅出的介绍。全书内容分为5个部分，对数据建模简介、数据模型要素，概念、逻辑和物理数据模型、数据模型质量以及数据建模的进阶内容等方面进行讲解，全面细致地为读者解答与数据建模相关的知识点和疑问。除此之外，本书的最后还对各类专业术语进行了细致的解释，方便读者参考。

本书是一本经典的数据建模指南，非常适合对数据建模感兴趣的读者以及从事数据科学等相关工作的专业人士参考阅读。

对本书的赞誉

对本书的赞誉

Steve Hoberman创作了一部内容丰富、生动、易于理解、实践性强的数据建模著作，而对于任何涉及信息技术领域的专业人士而言，数据建模无疑都是非常重要的。Steve Hoberman在本书中，清楚地回答了什么是数据建模、为什么会有数据建模，以及怎么进行数据建模等关键问题，并且通过适当的示例、类比和练习进一步强化了涉及的各个知识点。

——Len Silverston

畅销图书*The Data Model Resource Book*（卷1、卷2和卷3）的作者

数据建模作为有待探索且极具有潜在价值的领域，其商业价值往往隐藏于某个组织的信息技术部门。本书既强调了由此导致的商业价值的损失，也提出了如何体现其价值的措施。在“为什么”和“如何”进行数据建模方面，给出了一个易于理解和详尽的指导，同时也提醒我们IT项目开发的成功策略至少和所使用的信息技术同样重要。

——Chris Potts

企业IT策略师及畅销图书*Creating the Ultimate Corporate Strategy for Information Technology*的作者

对于想了解数据建模的初学者来说，本书无疑是一个非常好的参考指南。Steve Hoberman列出了数据建模的基础知识，并且用一种易

于理解又非常有趣的方式表现出来。我相信每位读者都能从中汲取到自己所需的内容。

——David Marco

EWSolutions公司总裁

非常好的一本书，读起来很有趣。Steve抓住了数据建模的精华并将其简化，对于不从事直接数据建模工作但又需要参与建模的读者而言，这是一本非常好的入门指南。对于偶尔进行数据建模的读者来说，这是一本非常有价值的参考书。对于具有丰富经验的建模者来说，这本书会时刻提醒你应该始终保持建模过程的简单化。

——David Wells

商业智能顾问及讲师

作为一名数据架构师和数据库设计者，我购买过很多本相关的图书。对于初学数据建模的技术人员和业务人员，本书是一个非常好的工具。Steve用自己的方式将数据建模的复杂性和基础知识进行讲解，无论读者具有怎样的经验层次和背景都能理解。如果想快速上手，本书将是读者的不二之选。我曾多次推荐本书，总能被多数人欣然接受。

——Tom Bilcze

Westfield 集团首席数据库设计师

本书是数据建模初学者以及想拥有“话语权”并想理解建模概念的人的必读之作。读者在阅读时，会有种作者陪伴左右的感觉，作者会向你逐一介绍各个术语，解释各个符号，告诉你动手之前、建模过程中以及建模结束之后应该考虑什么。

——Robert S. Seiner

总统KIK咨询及教育服务有限责任公司总裁

tdan.com数据管理简讯责任人

作为每天需要工作的数据架构师，有时甚至会忘记为什么进行数据建模。我只是知道了工作主题并按自己习惯的工作方式完成任务。我需要一个有用的定义，但有时候发现很难和其他人解释明白，我采用Steve的示例与他们交流，告诉他们我要做什么以及为什么这样做，令人高兴的是所有人都能明白。

——James Lee

健康服务数据架构、报表主管

这是一部近乎完美的图书，其内容覆盖面广，但同时又将所教授的内容保持在一个合理的水平，保证其简洁性和易用性。本书的可读性很强（我几次就读完了），将一个有效且易于理解的名片案例贯穿始终。

——Wayne Little

Creative数据解决方案公司CEO

致谢

在我的生命中有许多大咖（至今仍熠熠生辉），指引我前行。

这些从事数据管理行业的大咖有：UML领域专家Mickael Blaha；善于语言表达的Wayne Eckerson；对于数据建模富有极大热情（而且对我的第1版图书给出了中肯的评价和建议，并在第2版中做了相应修改）的David Hay；数据仓库领域的卓越贡献者以及对非结构化数据处理等未来趋势具有敏锐观察力的Bill Inmon；带来了元数据主流处理方法的Dave Marco；推动数据治理领域的发展，并发行了数据管理业界极具价值的刊物Tdan.com的Bob Seiner；引发如何建立数据模型的思考，并给出了如何提高团队合作的实践性技术的Graeme Simsion；多才多艺且广泛涉猎智能商业、数据建模、职业规划、PowerPoint、摄影、啤酒等领域的David Wells。

数据大咖们还通过像DAMA这样的用户组推动着数据管理领域的发展，通过志愿服务、个人按月或按季度组织学术讨论、安排大会发言、撰写报告等活动推动行业进步，并与各类从业者紧密相连。由于篇幅有限，在此列举出一些与我共事多年的数据大咖：Kasi

Anderson、Davida Berger、Tom Bilcze、Michael Brackett、Jimmy Chen、Susan Earley、Ben Ettlinger、Deborah Henderson、Jeff Lawyer、Carol Lehn、Wayne Little、Mark Mosley、Bill Nagel、Cathy Nolan、John Schley、Ivan Schotsmans和Anne Marie Smith.

还有其他人对这本书的出版给予了积极支持。感谢Bill Graeme和Michael对本书内容的补充，感谢Jeani对第1版的修订，感谢Carol出色的编辑工作，感谢Mark非常精彩的封面设计，感谢Abby完美的卡通设计。

当然还应该感谢那些数据世界以外的人们。感谢父亲的正直诚实、职业道德以及解决问题的能力。感谢母亲为我树立了一个热爱分享知识的榜样。感谢Jenn一直让我的生活很甜蜜。感谢Sadie和Jamie一直陪伴着我，并且提醒我让每天的生活简单化。

序言

序言

数据模型是构建应用系统的核心，是尽可能精准地表示业务运转的概念性框架。数据模型定义了操作者、行为以及管理业务处理流程的规则，并将定义内容用人们和应用程序都能理解的标准语法进行描述。本质上，数据模型将业务中涉及的概念转换为计算机代码，以致于应用程序和计算机系统都能按设计者的意图处理各类信息。如果没有数据模型，任何组织机构都不可能实现信息的自动化处理。

鉴于数据模型在应用系统开发过程中扮演着关键角色，毫无疑问，数据模型将决定应用系统开发及使用效率。即便程序设计方面已经做到了完美，但不良的数据模型设计同样会带来灾难性的破坏。执行性能下降，不精确的查询结果，没有弹性的规则和不一致的元数据等都是不良数据模型引发的后果。

另一方面，设计精良的数据模型是企业用户与信息技术专家之间的桥梁。在应用系统项目开发之初，借助数据模型企业与信息技术专家间就业务运转达成共识。信息技术专家将业务运转用概念数据模型及逻辑数据模型进行描述。企业用户则可以对模型进行审阅，在编写程序代码之前对模型进行必要的更正和改进。

很难想象有谁能像本书作者Steve Hoberman那样，用如此简单朴素的语言解释数据模型，很多数据模型工程师因此沉醉于他们的工作实践中。如果没有Steve，谁可能将Steve为The Data Warehousing Institute讲授的课程教得如此生动有趣，清晰明了？像在Steve所著的另

一本著作（*The Data Modeler's Workbench*）中看到的一样，Steve不仅知识渊博，而且还非常善于与各种读者沟通。Steve对于数据建模技术拥有无与伦比的热情和能量。同时，Steve还是我们研究中心里一位最受他人爱戴的成员之一。

符合庞大的需求。非常高兴Steve决定撰写这本著作，因为这类图书拥有巨大的市场需求。即使数据模型对于应用系统的开发至关重要，但仍有一大批业务人员和部分技术人员缺乏对数据模型的理解。这本著作的问世，无疑将唤起众多业务及技术人员对数据模型重要性的认识。

特别地，那些应用系统开发的倡议人，或被安排进项目组的业务人员，将发现这本著作是非常适宜的入门读物。对于刚刚入行进行应用系统设计的专业技术人员，这本著作同样是快捷、简单学习数据建模基础的优秀读物。大学教授为了帮助学生们掌握数据建模的有关概念、术语、成功准则等，这本著作也很值得推荐给他们。

—Wayne W. Eckerson

数据仓库研究服务中心主任

前言

相信很多读者和我一样，通常都会略过前言直接进入正文。但还是强烈推荐读者能先从前言部分开启本书之旅。前言将帮助读者对每一单元、每一章节有一个大体认识，并事先了解各部分的学习目标。

本书的10个目标

1．将会理解在什么情况下需要数据模型，以及各种情形下最适当的数据模型类型是什么。

2．能像阅读一本小说那样，轻松自如地读懂任何规模和复杂度的模型。

3．具备创建完整的规范化关系数据模型和维度模型的能力。

4．具备将一个逻辑模型转换为高效物理模型的能力。

5．具备使用模板工具，高效获取应用需求的能力。

6．具备解释数据模型记分卡中10个计分项的能力。

7．掌握如何与其他人员建立良好工作关系的实践经验。

8．了解非结构化数据及其模型化。

9．了解UML的基本概念。

10．具备XML环境中创建数据模型的能力，并了解元数据和敏捷开关的基本概念。

本书包含有5个部分，第1部分引入数据建模，并介绍了数据建模的目的和变化。第2部分说明数据模型中的所有组件。第3部分介绍关系型和维度型概念模型、逻辑模型和物理模型。第4部分则关注如何使用模板提高数据模型质量，介绍数据模型记分卡以及如何与业务人员、项目团队高效沟通。第5部分讨论关于数据建模的常见疑问。

将本书内容与10个学习目标关联起来看，第1部分的前半节完成了目标1，第2部分完成了目标2，第3部分完成了目标3和4，第4部分完成了目标5、6和7，第5部分则完成目标8、9和10。

第1部分由3章组成。第1章引入数据模型，并通过两个实例（冰淇淋和名片）说明数据模型的作用，这两个实例贯穿始终，便于读者对需求分析到模型设计的整个建模过程有所认识。第2章介绍了数据模型的两个非常有价值的特征：交流性和精确性。同时本章还就数据模型最行之有效的领域给予讨论。第3章将数据模型与照相机做以类比，说明关于照相机的4种设置同样适用于数据模型。理解4种设置对数据模型的影响将极大增加建模成功的可能性。（注：应用系统是为特定用

户设计的以实现一定功能的一个程序或程序集，如文字处理系统、订单处理系统、利润报表系统等。）

第2部分包含随后的4章，用以介绍数据模型组件。第4章介绍实体，第5章介绍属性，第6章介绍关系，第7章介绍键。

第3部分由随后的3章构成，其中讨论了概念模型、逻辑模型和物理模型这3种不同类型的模型。第8章着重学习概念模型并讨论了在创建概念模型过程中的3种变化。第9章学习关系及维度逻辑模型。第10章介绍物理模型，重点学习使用反规范化和分区等不同技术实现物理模型的高效设计，同时还将学习渐变维度模型。

第4部分包含3章内容。讲解如何使用模板、数据模型记分卡及如何有效地与业务人员、项目组成员进行交流沟通，从而提高数据模型质量。第11章推荐了多种用于获取、验证用户需求的模板，模板的使用将有助于降低时间开销并提高建模精度。第12章讲解数据模型记分卡以验证数据模型质量。第13章介绍了如何与其他团队成员协作以及高效共事的一些实践经验。

第5部分也包含3章内容，其中介绍了凌驾于数据建模之上的有关主题。第14章介绍非结构化数据，因为非结构化数据的处理是当前流行的趋势。本章介绍了分类、本体两个处理技术。第15章学习统一建模语言UML中涉及数据模型的内容。第16章给出了经常被提及的5个疑问，并一一解答，其中包括XML、元数据、敏捷开发。

第2版在第1版的基础上做了很大的改进。所有章节相比第1版都变化很多，其中更多地引入了新技术和示例。而且第2版更注重数据模型

创建过程。作为强化概念，关键点都被添加至每章的结尾。每章开篇之处也添加了3行新体诗，给出了各章梗概。

本书还引入一则新术语：路径搜寻（Wayfinding），并重点介绍了如“元数据”等多个建模领域中容易被混淆的概念。本书还添加一些很有针对性的习题，并给出了参考答案。本书最后还罗列出本书涉及的全部名词解释。

本书的另一大特色在于其并非由一名作者独立完成。在写作之初，我曾尝试撰写有关UML和非结构化数据有关的内容，但我很快意识到其他专家学者的作品更好。于是请Graeme Simsion、Bill Inmon和Michael Blaha这3位专家分别撰写了本书的第13章、第14章和第15章。

数据建模不只是一种工作或职业，它还是一种思想，一种无价的过程和生活方式。但请尽量保持其简单实用，现在一起开始建模之旅吧。

第1部分　数据建模简介

第1部分将引入数据建模，并介绍了数据模型的目的及其类型。完成该部分学习之后，读者将可以对在什么情况下需要引入数据模型进行判断，并可以根据实际情况选择适当的数据模型类型。读者还应该可以通过数据模型特征进行模型评估，并能针对特定的模型确定其特征的优劣及确定该模型与其创建目的是否吻合。

第1章将引入数据模型，并通过两个实例对这一强有力工具进行阐述。这两个实例也将贯穿整本教程。因为我个人偏好甜品，所以一个实例与冰淇淋有关（是的，冰淇淋）。另外一个实例是对名片进行数据建模。无论是冰淇淋，还是名片，都用来说明建模技术，这样读者可以从需求分析到模型设计了解整个建模过程。

第 2 章介绍了数据模型的两个非常有价值的特征：交流性和精确性。读者将了解到模型交流性如何体现以及 3 种可能弱化模型精确性的情形。本章还从业务及应用程序两个领域对数据模型的应用进行了说明。

第3章将数据模型与照相机进行比对，说明用于照相机的4种设置可以完美适用于数据模型。对数据模型设置的理解将极大增加应用程序开发成功的可能性。本章还比对了图像格式与数据模型，由此引入数据模型的3个层次：概念、逻辑、物理。

第1章　数据模型

我怎样才能到达目的地?

地图、设计蓝图、数据模型

请为我指引迷津。

当我又一次意识到自己完全迷路的时候，我懊恼地重重拍了一下方向盘。要知道，我正独自行驶在法国的公路上，赶着去参加一个非常重要的商务会议，而且此时距离天亮还有一个小时，还好我发现前方有一家正准备开张的加油站，我停下来，走了进去，并把目的地的地址拿出来给服务员看。

我不会说法语，那个服务员也不会讲英语，我需要帮助，但无法通过言语交流，幸亏他认出了我要访问的公司的名字，最后他拿出了纸笔，给我画了一张示意图。如图1.1所示，他用线条表示街道，用圆圈表示环岛路口并配有相应的数字表示出口，还用矩形框表示加油站（Petrol）和我的目的地（MFoods）。

这个由服务员绘制的地图里，只包含与我的行程相关的信息，在它的帮助下，我顺利抵达目的地。事实上，这张地图就是一个我旅行所需要的实际道路的模型。

地图是对复杂地理景观（geographic landscape）的简化，同理，数据模型也是对复杂信息景观（information landscape）的简化，本章将以冰淇淋和名片为例，介绍被誉为路径搜寻工具（wayfinding tool）的数据模型及其重要作用。

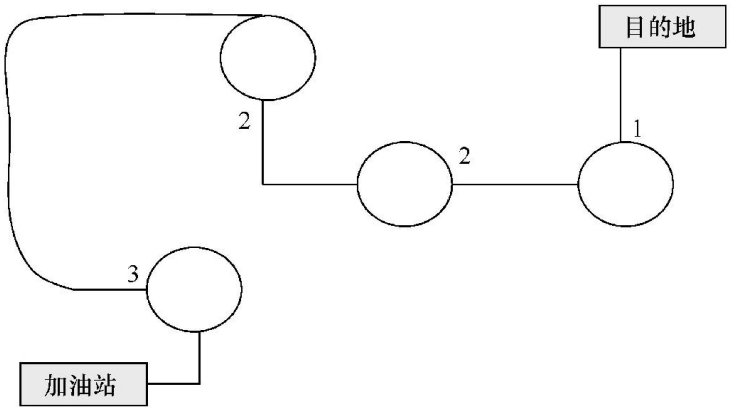

图1.1　简单的地理景观示意图

1.1 路径搜寻说明

如果“数据模型”不能引起你或你的业务伙伴们的兴趣，你可以使用“路径搜寻（wayfinding）”予以替代，路径搜寻囊括所有被人类或动物使用的技术及工具，以实现从一个地点抵达到另外一个。如果一位旅行者用天空中的星斗导航，那么星斗便是他的路径搜寻工具，同理，地图、指南针也都是此类工具。

所有的模型也是路径搜寻工具。模型是一组文字及各类符号的集合，用来将一个复杂的概念简单化。我们生活在一个令人眼花缭乱的世界，人们很难将注意力集中在一些关键信息上，从而无法做出一个明智的决策。而地图可以帮助旅行者游览一座城市，组织结构图可以帮助员工理解组织间的相互关系，设计蓝图则可以帮助建筑师交流建造计划。所以，地图、组织结构图、设计蓝图都是对复杂事物的过滤和简化，以帮助人们理解现实世界，提高路径搜寻能力。

在法国的这次旅行，要不是加油站服务员绘制了地图，让我立刻明白如何抵达目的地，我可能得多花几个小时，并且不断碰壁。模型则使用一些标准符号让人们快速地理解相应的内容。例如，在服务员绘制的地图里，他用线条表示街道，用圆圈表示环岛路口，正是这些符号帮助我在脑海中映射出一条条街道和一个个路口。

1.2 数据模型说明

当我还在读大学的时候，课堂上教授们经常会在挂图板上写下大量内容，而学生们则疲于整理笔记。在这种情况下，“信息过载”（information overload）可以用来形容这种状况，即当前的信息量超出了大脑所能接受的最大信息量。此时最好在校园里闲逛一会，亦或打打网球，亦或玩半小时的太空入侵者游戏（Space Invaders），让身心得以放松，以便接受更多信息。然而现代社会，人们创造并接受越来越多的信息，但休息、放松的时间却越来越少。而且我经常听到这样的说法—在世界范围内，信息量以每年60%的速度递增，这让我不禁感叹，在如此众多的信息面前，我们真正掌握、理解的信息是多么有限。

幸运的是，数据模型这一工具可以帮助我们有效地简化所有信息。类似于路径搜寻工具，无论是商务专员，还是IT专家，都可以有效地使用数据模型，即利用一组符号、文本来准确表达真实信息的精简子集，以便改善某一组织内部的交流、沟通，并提供一个更灵活、更健壮的应用环境。例如，在法国地图上用线条表示公路。又如，在数据模型里可以把“客户”这两个字用矩形框起来，表示一些实际、具体的客户，如Bob、IBM、Walmart。

换言之，地图是对复杂地理景观的简化，而数据模型则是对复杂信息景观的简化。很多情形下，现实数据的极其复杂性使得数据模型看起来异常简单，例如服务员给我绘制的那些环岛路口。

数据模型是一组由符号、文本组成的集合，用以准确表达信息景观，达到有效交流、沟通的目的。描述信息景观的方式多种多样，本书主要使用矩形框、线段等元素描述数据模型，当然还可以使用统一建模语言（UML）类图（Class Diagrams）、电子表格（spreadsheets）、状态转换图（State Transition Diagrams）。所有这些模型都可以视为在复杂信息世界里的路径搜寻工具，都可以显示对复杂信息世界的简化。

1.3 有趣的冰淇淋

电子表格可能是我们在日常工作生活中最熟悉的一种数据模型。电子表格是纸质工作表格的一种表示形式，表单中包含由行和列构成的网格，网格中的每个单元格都可以存放文本或数字，表单中的列通常表示不同类型的信息。假设我刚刚结束一段旅程返回罗马，我喜欢那里的冰淇淋（gelato），当我们一起走进一个冰淇淋店时，你应该会注意到几个表单，表1.1为一个冰淇淋口味列表，表1.2则包含了冰淇淋大小及价格信息。

表1.1 冰淇淋口味

香蕉
卡布奇诺
巧克力
巧克力片
咖啡
猕猴桃
软糖

香蕉
开心果
草莓
香草

表1.2 冰淇淋大小及价格

1匙1.75
2匙2.25
3匙2.60

上述表单也是一个数据模型，因为它用一组符号集合（本例中用的是文本）来描述现实世界的一些事物（本例中描述了美味的冰淇淋口味及其价格）。你们猜猜我买了几匙巧克力口味的冰淇淋?

数据模式形式（data model format）是本书的主题之一，而且与上例中的表单非常类似。虽然数据模型是一个较宽泛的概念，但这里需要注意的是当使用数据模型这一术语时，其形式需引起我们足够的重视。但不同于数据表单，数据模型应满足如下要求。

- **只包含类型：**数据模型中通常无需显示，如巧克力或3匙，这样具体的数据，需要显示的是数据对应的概念或类型。比如，上述数据模型中显示的类型为冰淇淋口味，而非巧克力或香草这样具体的值，还显示了冰淇淋大小，而不是具体的值，1匙或2匙。
- **包含相互作用：**数据模型还需要抓住不同概念、类型间的相互作用。比如，冰淇淋口味与大小之间的相互作用是什么？如果有人要买3匙冰淇淋，那么这3匙是同一种口味，还是3种不同的口味。正如冰淇淋口味与大小间的相互作用，在一个数据模型中要求表述不同类型间的相互作用。
- **提供一个简洁的交流媒介：**比起仅使用数据表单进行交流，用包含数据模型的文档交流，其效率要高得多。数据模型显示各个类型，并用简单且有效的符号表达它们之间的相互作用。对于冰淇淋这个实例，为了有效描述各个类型以及它们之间的相互作用，显然数据模型是种更为精练的工具，而仅使用数据表单往往达不到这样的效果。

1.4 有趣的名片

名片（Business Card）包含了丰富的关于某人及其单位的信息。本书中，我会用名片作为基本模型，来阐述许多与数据模型相关的概念，通过构建一个名片数据模型，我们可以亲身感受到从具体的名片上能获得多少信息，或者从更广泛意义上的联系人管理领域能获得多少信息。

我打开床头柜抽屉（惊人的是自从20世纪90年代中期抽屉就未被整理过），抓起一把名片，铺在桌上，挑出最有趣的4张建模。第1张是我本人现在的名片。第2张是多年以前妻子和我创办的互联网公司的名片。还有一张是一位魔术师的名片，他曾经在我们的聚会上表演过。最后一张是我最钟爱的一家饭店的名片。为了保护个人隐私，我修改了姓名和联系方式，如图1.2所示。

图1.2　床头柜里的4张名片

在这些名片上你能看到什么信息?

假设我们这次练习的目的是理解名片上的信息，并以实现一个成功的联系人管理应用程序为最终的目标。让我们先列出以下一些信息。

Steve Hoberman & Associates, LLC

BILL SMITH

Jon Smith

212-555-1212

MAGIC FOR ALL OCCASIONS

Steve and Jenn

58 Church Avenue

FINE FRESH SEAFOOD

President

我们很快就能意识到，尽管这里只处理4张名片，但是即便列出所有的信息，对于帮助理解数据模型也是非常有限的。进一步地，设想一下如果我们要处理的名片不仅仅局限于4张，而是扩大到床头柜里的所有名片，或者更糟，扩大到曾经收到的每一张名片！很快，数据量就超负荷了。

数据模型将数据汇总，从而让它们更容易理解。例如，我们查看下列数据，发现这组数据适合放在一个被命名为“公司名称”（Company Name）的数据组中（电子表格中的列标题）。

Steve Hoberman & Associates, LLC

The Amazing Rolando

findsonline.com

Raritan River Club

另外一个电子表格中的列标题应该为“电话号码”（Phone Number）。表1.3为一个列出部分名片信息的表单。

表1.3 名片信息

	公 司 名	电 话 号 码
名片1	Steve Hoberman & Associates, LLC	212-555-1212
名片2	findsonline.com	(973) 555-1212
名片3	The Amazing Rolando	732-555-1212
名片4	Raritan River Club	(908) 333-1212 (908) 555-1212 554-1212

再进一步做这个练习，我们可以将名片中的不同数据组织到以下各个组中。

姓名Person name

职务Person title

公司名称Company name

电子邮箱Email address

网页Web address

通信地址Mailing address

电话号码Phone number

标志Logo (the image on the card)

专业Specialties (such as “MAGIC FOR ALL OCCASIONS”)

至此，结束了吗？这组列表就是一个数据模型？答案是否定的。我们丢失了一个关键要素：数据组之间的相互作用或关系。例如，公司名称和电话号码之间有什么关系？一个公司可以有多个电话号码吗？一个电话号码可以属于多个公司吗？没有电话号码，一个公司可以存在吗？在建立数据模型的过程中，这一类问题都需要被提出并解答。

为了建立任何一种路径搜寻工具，人们通常在迷路足够多次之后，才有可能发现正确的路径，例如第一个为某地区绘制地图的人，一定会花费很多时间，走过很多弯路，才能完成其工作。可见绘制地图是一个具有挑战性并需要一定时间花销的过程。

创建并完成一个数据模型往往会遇到相同的情形，与概念“数据模型”相应地还有一个概念“数据建模”。数据建模是建立数据模型的过程，更具体地说，数据建模为了明确某一组织结构及其操作，而使用一组技术和实施一些活动，即提出一个信息解决方案，从而实现该组织的某些目标。当然在数据建模过程中，还需要很多技能，如专心聆听，尽可能提出大量问题，甚至耐心。

数据建模者要求能与来自不同部门，具有不同技术背景，不同业务经验，不同技能水平的人员交流、沟通。在交流中，数据建模者不

仅需要理解每个人员的观点，而且还需要通过反馈证明理解无误，最终作为组件，构建在模型中。在一个项目的初期，通常数据建模者没必要去处理所有数据模型所需的数据，但阅读大量相关文档、咨询数百个与业务有关的问题则是必要的。

1.5 练习1：教教你的邻居

为了强化数据模型认识，读者可以试图向非IT人士，如邻居、家人或朋友，解释这一概念。

他们听懂了吗?

在本书的后面有关于如何解释数据模型这一概念的参考答案。

> **关键点**
>
> √ 路径搜寻囊括所有被人类或动物使用的技术及工具，以实现从一个地点抵达到另外一个地点。
>
> √ 数据模型是一组由符号、文本组成的集合，用以准确表达信息景观，达到有效交流、沟通的目的。
>
> √ 数据模型具有多种表现形式，而最常见并得到广泛理解的形式为电子表格。
>
> √ 数据模型形式是本书的主题之一，它与电子表格非常相似，但数据模型基于类型，包含相互作用和可扩展性。
>
> √ 数据建模是建立数据模型的过程，需要很多与技术无关的技能，如专心聆听，尽可能提出大量问题，甚至耐心。

第2章 为什么需要数据模型

笼统地讲

数据模型是精确的

0，1……还是很多。

数据建模是构建应用程序的必要组成部分。数据模型之所以如此重要，是因为它所带来的两大核心价值—交流性及精确性。数据模型可以有效应用于业务及应用程序开发领域，本章则通过讲述数据模型在这两个领域的使用，阐明数据模型的两大核心价值，你将学习到数据模型对交流的促进作用和能削弱数据模型精确性的3种情形。

2.1 交流性

来自不同部门、职能区域，以具有不同技术背景和业务经验的各类人员时常需要就业务问题进行讨论并最终做出决策。讨论中，需要明确对方对诸如“客户”“销售”等这类概念的观点。数据模型作为一种理想的工具，可以有效达到理解、记录并最终协调不同观点的目的。

当我身在异国，无法进行言语交流时，那位加油站服务员为我绘制的地图模型，使我明确了如何抵达目的地。无论我们想尝试着去了解某一业务中的一些重要概念如何与其他概念相关联，还是想了解一个已经使用了近20年的订单处理系统的运作，数据模型都是一个用于解释信息的理想工具。

借助数据模型，我们可以在不同的细节水平上交流相同的信息。例如，前不久我们构建了一个用于描述快餐领域消费者间相互作用和影响的高层次数据模型。于是，当有消费者电话投诉公司产品时，我们所构建的模型将存储该投诉以及与其相关的信息。可以看出在这个项目中，那些重要的商务客户就与我们建立的这个高层次数据模型所展示的内容相关联。数据模型有助于限定项目范围，帮助理解诸如客户、产品及相互作用等关键观念，帮助建立融洽的业务关系。几个月之后，我们使用更细化的模型来描述消费者间的相互作用信息（consumer-interaction information），并向业务报表制作者说明，在每一种选择条件下，哪些信息将出现在报表中。

基于数据建模的交流，并非只是在建模结束后才开始的。事实上，伴随着数据建模进程，需要更多的交流和知识分享，即交流沟通在建模中与建模后都同样具有价值。下面让我们一起领略建模过程和建模结果所带来的交流价值的更多细节。

2.1.1 建模过程中的交流

在建立数据模型的过程中，我们必须分析数据及数据间的关系，我们别无选择，必须对所要模型化的内容具有清晰的认识。人们在建模过程中，相互挑战、质疑，从而获得与术语、假设、规则和概念相关的大量知识。

在为一家大型制造业公司建立配方管理系统（recipe management system）数据模型的过程中，我惊讶地目睹了具有多年工作经验的项目组成员就“组件”（Ingredient）的概念和“原材料”（Raw Material）的概念是否存在差别进行辩论，经过30分钟有关组件与原材料的讨论，每一位参加建模的人员都从中受益，当结束建模会话（modeling session）时，他们都对配方管理有了更深入的理解。又如，以模型化名片为例，在建模过程中，将学习到许多有关人员、公司和联系人管理的共识。

2.1.2 建模过程后的交流

创建并完成的数据模型是讨论在应用程序中哪些模块应该被构建的基础，甚至更底层的，借以数据模型讨论业务流程或程序功能模块如何运作。数据模型像一张可反复使用的地图，无论是分析师、建模者，还是开发者，都可以利用它，了解他们各自关心的对象如何工

作，正如第1位地图制作师需要经历艰苦的学习，才能准确记录下地理景观，为他人导航。与此极其相似的是建模者也需要经历类似的训练（痛苦但却有益）以便让其他人能够理解一个信息景观（information landscape）。

当我准备进入一家大型制造业公司工作之前，我的新任主管给了我一本公司手册，其中记录了一组与公司有关的数据模型，当我阅读了好几遍之后，我已经对公司业务中的重要概念和业务规程相当熟悉了。所以，在我工作的第一天，我已经掌握了大量关于公司业务运作的信息，甚至当同事们提及一些专有术语的时候，我也能熟知它们的含义。

就上一章提到的名片，一旦完成相应的数据模型，其他人就可以通过该模型了解联系人管理了。

2.2 精确性

数据建模的精确性指的是阅读模型时，其中的每一个符号和条目都是清晰、无二义性的。你可能与其他人争议所使用的规则是否准确，但这与我们所强调的模型的精确性是不一样的概念。换言之，如果你看到模型中的某一符号并说“我看见了A”，那么另外一个看到这一符号的人不可能说“我看见了B”。

再回到那个名片的例子，假设我们定义“联系人”为名片上所罗列的人或公司，或许有人提出“一个联系人有多个电话号码”。显然这个表述是不精确的，因为我们不确定一个联系人是否可以没有电话号码，或者必须有一个电话号码，或者必须有多个电话号码。类似地，我们不明确是否允许出现一个未与任何联系人关联的电话号码，或者一个电话号码必须属于某一位联系人，或者可以属于多位联系人。数据模型提出的精确性，要求将这些模糊的表述转换为以下断言。

- 每一位联系人必须和一个或多个电话号码关联。
- 每一个电话号码必须属于一位联系人。

由于数据模型引入了精确性，所以无需试图花费宝贵的时间来解释模型，相反，时间可以用来讨论、验证一些与建立某一模型相关的概念。

但是在3种情况下，数据模型的精确性可能降低。

1．弱定义：如果对一个数据模型中的一些条目（terms）的定义，缺乏根据或压根不存在，那么此时极有可能对这些条目产生多种理解。如果数据模型中的一则业务规则规定每一位雇员（Employee）必须拥有一套福利计划，同时又将“雇员”定义为碳基生物形式这样一种缺乏现实意义的表述，那么我可能认为“雇员”包括“工作申请人”，而你可能认为不包括“工作申请人（Job Applications）”，所以你我之间必将有一位是错误的。

2．伪数据：第2种情形出现在当某一数据超出了常规的取值，而我们又希望将其引入特定的数据记录中。一个绕开数据模型严谨性（rigor of data model）的老把戏是扩大数据模型可能包含的数据值。例如，出于某种考虑，要求联系人必须有至少一个电话号码，而如果要添加到应用程序的联系人并没有电话号码，那么某位程序使用者可能为该联系人创建诸如“不可用”“99”或其他假电话号码，该联系人最终被添加进了应用程序。这个例子中，使用伪数据将一位没有电话号码的联系人添加进了应用，从而违背并规避了我们最初制定的业务规则。

3．模糊或缺失的标签：阅读一个数据模型类似于阅读一本书，应该有正确的句子结构，动词是句子中非常重要的组成部分。对于数据模型，这些动词用来描述模型中一些概念间的相互关联。以“客户（Customer）”和“订单（Order）”这组概念为例，可以通过动词“订购”（place）把它们相互关联，即“一位客户可能会订购一个或多个订单”。而诸如“联系”“有”等模糊的动词，或者缺少动词，将降低整个数据模型的精确性，正如我们不能准确理解一个句子的含义一样。

数据模型的精确性还源于使用了一组标准的符号集合，那家加油站服务员为我绘制的交通图使用了标准符号，于是人人都能理解。我们马上就会学到一些数据模型中使用的标准符号。

2.3 使用数据模型

从传统的角度来讲，不仅要求对一个新的应用进行不断的分析与设计，以明确所有满足该项目的必备条件，还应该对现有数据库具有完整、正确的认识，并在此基础上完成数据模型的构建。由于模型的精确性，数据模型还可以被用于以下几种情况。

理解已有应用程序。数据模型提供了一个简单而精确的视角，用来观察某个应用程序所涉及的概念。我们可以通过考察一个现有应用程序的数据库，并根据该数据库结构创建出一个数据模型。“逆向工程”（reverse engineering）这一专业术语，即表示根据现有的应用构建出数据模型的过程。不久前，一家制造业机构需要将一个已使用了25年的应用系统迁移到一个新的数据库平台，对于这个庞大的应用系统，为了掌握理解它的结构，我们将数据库逆向工程为一个数据模型。

风险管理。通过数据模型可以获取一些概念及概念间的相互作用，并且这些概念及相互作用受到程序、项目开发的影响。对一个现有应用程序进行结构性添加或修改将产生什么影响？有多少应用程序结构需要备份？现在有很多机构购买一个软件后会再对其进行自定义修改。影响分析（impact analysis）是进行风险管理的一种方法，借助数据模型进行影响分析，来明确对所购买的软件进行结构修改会产生什么影响。

了解业务。开展一个大型项目开发的必要条件是在了解应用程序如何辅助业务开展之前，你最好先去了解相关的业务流程。例如，在开发订单录入系统之前，得先了解订单录入的处理过程。我最欣赏的一句话源自威廉·肯特（William Kent）1978年所写的一篇名为“数据与实现”（Data and Reality）的文章，文中当肯特论述到创建一个数据库来存储图书信息所需要的步骤时，他写到：所以需要再次强调的是如果计划创建一个图书数据库，在还未了解某个概念的准确含义之前，最好在所有用户中达成共识，如什么是“一本书”。

培训团队成员。当新成员想要尽快跟上进度或开发者想要了解需求时，数据模型可以作为一个非常有效的阐述工具。一位新人无论何时加入我们的部门，我都会花费一些时间，通过一系列数据模型尽可能快地给他传授一些相关概念。

2.4 练习2：转变非信仰者

在你所在的组织中找到一位数据模型的非信仰者，并试图转变他。你都碰到了哪些障碍？你是否说服了他们？

> **关键点**
>
> √ 数据建模的两大核心价值是交流性及精确性。
>
> √ 无论是建模中，还是建模完成后，都需要进行交流、沟通。
>
> √ 如果存在弱定义、伪数据、模糊或缺失标签等3种情况，数据模型的精确性将会降低。
>
> √ 交流性和精确性使得数据模型成为一种构建应用程序的出色工具。
>
> √ 数据模型还可以被应用于理解已有应用程序、了解业务、执行影响分析和培训团队成员。

第3章 哪些相机设置也适用于数据模型

相机设置

变焦、对焦、定时器、滤镜

数据模型也一样。

本章将数据模型与相机比较，解析4种相机上的设置，它们完美诠释了数据模型，理解这些设置对数据模型的影响，将有助于增加一个应用项目成功的几率。同时，本章还对比了3个层次上的图像格式，从而理解概念模型、逻辑模型和物理模型。

3.1 数据模型与照相机

一个相机上可以使用很多设置，来确保拍出动人的画面。想象一下，你正用相机瞄准一个美丽的落日场景，即使面对同一场景，如果使用不同的对焦、定时器或变焦设置，那么你可能也会拍到完全不同的照片。例如，你可以推远镜头以捕获尽可能多的落日画面，还可以拉近镜头，将画面集中在一位在落日中漫步的游客的身上，这完全取决于你想要将什么呈现在照片中。

变焦、对焦、定时器、滤镜是与相机有关的4种设置，它们都可以被直接变换到数据模型上，如图3.1所示，每种相机设置都对应于一个数据模型的特征。

图3.1　相机设置向数据模型的变化

通过变焦设定，可以允许摄影者捕获一个广阔的场景而忽略一些小细节，或者捕获一个强调细节的狭窄范围。类似地，对数据模型的范围（scope）设置可以改变一个数据模型所能呈现的信息量大小。相机的对焦设置可以决定照片中的景物是锐化的（sharp），还是模糊的（blurry）。类似地，对模型的抽象（abstract）设置则可以使用诸如同类（party）、事件（event）等通用概念来“模糊”（blur）概念间的区别。定时器可以用来设定一个实时快门，或一段时间之后的快门。类似地，对数据模型的时间（time）设置则可以用来获取一个当前的视角或未来一段时间后的视角。滤镜设置可以用来调整整个画面的外

观，产生某种特定的视觉效果。类似地，数据模型的功能（function）设置则可以用来将模型调整到业务视觉或应用程序视角。

同时，不能忽略图像类型的重要性。摄影校样（proof sheet）允许在一张纸上展示所有的图像，而底片为Raw格式的图像，其可以输出很多种图像格式，包括胶片、幻灯片或数字图像。类似地，相同的信息图像（information image）能够存在于数据模型的概念、逻辑、物理等3个不同的细节层次上。

哪种设置适合于你的模型？正如落日下的摄影，这取决于你想要捕获什么。用适当的模型设置匹配你的模型目标，可以提升数据模型以及它所支撑的应用项目的质量。

3.2 范围

数据模型和相片都有相应的边界，边界决定了能够被显示的事物。一张照片可以捕捉到我的小女儿正享受冰淇淋时的情景（实际上，她的整个面部都在享受着冰淇淋），或者可以捕捉到我女儿及其所处的环境，如冰淇淋店。类似地，数据模型可以只包含索赔过程（claims processing），或者还可以囊括所有保险业务中概念。典型的情况下，数据模型范围可以是一个部门、一个组织或一个行业。

- **部门（工程）**。最常见的建模任务类型是工程级范围（project-level scope），工程是完成软件开发任务的计划，经常由一组在指定日期之前可交付的成果所定义。例如，可以包括销售数据集市（sales data mart）、经纪人交易应用（broker trading application）、预定系统（reservation system）及对现有应用的加强。
- **组织（应用程序）**。应用是一种大型的、集中组织的计划，其中可能包含多个工程。通常应用具有起始日期，但如果成功，则没有结束日期。应用可能是非常复杂且需要长期模型化的任务。例如，可以包括数据仓库（data warehouse）、操作数据存储（operational data store）及客户关系管理系统（customer relationship management system）。
- **行业**。一份行业计划被设计，旨在获取行业中的一切，如制造业或银行业。很多行业都在进行大量的工作，致力于共享一个共用的数据模型。如健康卫生和电信等行业联盟，都在从事共用数据

模型结构的开发，这类共用结构可以加速应用程序开发以及方便同行业中不同组织间的信息共享。

3.3 抽象

一副照片可以是模糊或清晰的。类似于如何对照相机进行对焦，使得图片变得锐化或模糊，模型的抽象设置允许你表现“锐化”（concrete具体）或“模糊”（generic通用）的概念。

通过重定义和对模型中的一些属性、实体、关系进行合并，得到一些通用的概念，这样为数据模型带来一定的灵活性。抽象是指去除部分细节而保留一些重要的属性、概念或主题的必要本质，从而扩展适用性，满足更宽泛的应用需求。通过去除细节，消除分歧，改变我们看待这些概念或主题的方式，此时我们或许可以看到那些之前不太明显，甚至未曾发现的东西。例如，可以将“员工”“顾客”抽象为一个更通用的“人”的概念，人可以担任不同的角色，员工、顾客只是其中的两种，更多的数据模型抽象能将该模型变得更宽泛、通用。对于数据模型，概念可以被不同层次地抽象：“业务云”“数据库云”或“地面上”。

- **在业务云中**。在这一级别的抽象中，只有通用的概念被应用于数据模型，业务云模型通过使用诸如人（Person）、交易（Transaction）和文档（Document）等通用概念，隐藏许多现实复杂性。实际上，当使用业务云的概念时，糖果公司和保险公司变得非常相似，倘若你缺乏对某一业务的认识，或不能获取到一些业务文档和资料，一个业务云中的模型将能很好地运作起来。
- **在数据库云中**。在这一级别的抽象中，只有通用的数据库（database，DB）概念被应用于数据模型。数据库模型是最容易

被创建的，它使用诸如实体（Entity）、对象（Object）和属性（Attribute）等数据库概念。如果你不清楚业务如何开展，而又想要覆盖所有行业的所有领域，那么一个数据库云中的模型将能很好地运作起来。

- **在地面上**。这类模型对应于少量的业务处理，并使用尽可能少的数据库云实体，而使用大量能代表具体业务术语的概念。比如数据模型得花费大量时间来创建学生、课程、教师等3个概念，并允许增加一些具体的值来帮助理解业务处理、解决数据问题。

3.4 时间

大部分照相机具有定时器功能，使得摄像者可以在设定定时器后，快跑并把他自己也拍摄进画面中。类似于应用照相机定时器可以拍摄一幅当前或一段时间之后的场景，数据模型的时间设置允许将一个当前或未来的视角表现在模型上。

一个数据模型可以表示当前的业务运转，也可以表示未来一段时间后可能的业务状况。

- **当前**。一个带有当前设置的模型可以获取当前业务运作的信息。即便存在一些陈旧的业务规则，它们也得出现在模型中，即使在不久的将来这些规则要被修改。另外，如果一家企业正计划购买另一家公司，或出售一家公司，或者正在改变经营种类，那么当前视图也不会显示任何一个上述正要发生的变化，而仅仅只能表现出目前的状况。
- **未来**。一个带有未来设置的模型可以表现未来任意一个时间阶段的业务。通常这种模型是一个理想状态下的视角，无论过去了1年、5年，还是10年，未来设置总能体现该组织的发展方向。如果一个模型需要支持某个组织的发展规划和战略布局，那么设定一个未来设置将是其首选。我曾经作为负责人为一所大学构建模型，由于有大量的应用迁移要在一年内完成，所以这个模型需要表现出一年以后的情况。还需注意的是对于大部分组织，如果需要一个未来的视角，通常必须首先创建一个当前的视角作为起始点，这样做没有什么不妥，正如一位摄影者可以对一个场景拍摄

多幅照片，那么一位数据模型的创建者也可以用不同的设置去创建多个模型。

3.5 功能

滤镜是一组覆盖在相机镜头上的塑料和玻璃材质的滤光片，可以用不同颜色的滤光片对照片进行调整，例如，让照片看起来更蓝或更绿，与相机滤镜可以改变场景的外观一样，数据模型的功能设置则允许一个数据模型表现为业务视角或功能视角。我们正在模型化一个业务视角下的世界，还是应用程序视角下的世界？有时它们一致，但有时它们有很大的差别。

- **业务**。这种过滤器使用的是业务术语及规则，而模型呈现与应用无关的视角，无论某一机构是用文件柜存储信息，还是使用最有效的软件系统。在模型中，这些信息将会被一些业务概念表示。
- **应用程序**。这种过滤器使用的是应用程序术语及规则，是用应用程序的观点看待业务运作而形成的视角。例如，应用程序使用术语“对象”来表示“产品”，则产品会以“对象”的形式出现在模型中，而且是以应用程序定义术语的方式进行定义，而不是用业务处理的方式进行定义的。

3.6 格式

正如一台照相机可以用多种不同的格式获取图像，数据模型的格式设置可以用来调整模型的细节水平，让模型呈现出很宽泛、高层次的概念视图（conceptual view）或呈现出能反映更多细节的逻辑或物理视图（logical or physical view）。

- **概念视图**。通常当一组照片被冲洗时，一份校样会包含每一幅照片的缩略图，则观察者可以用一张相纸得到一个全景的视角，这里的全景视角类似于概念数据模型（conceptual data model，CDM）。概念数据模型可以在一个很高的层次上表示业务，这种很宽泛的视图仅包含给定范围内的一些基本、关键的概念。这里的“基本”意味着在一天的交谈中一些概念会被很多次地提及。“关键”意味着倘若没有这些概念，部门、公司、行业会被极大地改变。有的概念是所有组织通用的，如“顾客”“产品”和“员工”，而有的概念则特定于某一行业或部门，如保险领域中的“政策”，或中介行业中的“交易”。
- **逻辑视图**。在数码相机问世之前，一卷冲洗过的胶片可以得到一组底片，这些底片可以用来很好地观察所拍相片，这里底片类似于逻辑数据模型（logical data model，LDM）。逻辑数据模型描述了一份详细的业务解决方案，这使得建模者不用创建与软硬件实现有关的复杂数据模型，就能掌握相应的业务需求。
- **物理视图**。虽然底片是一种很好的观察相片的视角，但它其实并不实用。例如，你不太可能将底片置于相框或相册中拿去与朋友

分享，你应该转换或“实例化”（instantiate）底片为照片、幻灯片或数字图像。相似的，逻辑数据模型需要被修改成更实用的物理数据模型（physical data model，PDM）。它是逻辑数据模型的化身（incarnation）或实例化（instantiate），类似于照片是底片的化身，物理数据模型表示详细的技术解决方案，是对特定环境的优化（诸如特定的软件或硬件环境）。物理数据模型是在某种特定环境下，对逻辑模型执行力的修改、增强，在该环境中数据将被创建、维护和访问。

3.7 练习3：选择正确的设置

在下列列表中，为每种情形选出最适当的设置，参考答案在书的后面。

1．给一位项目组开发人员解释现存的联系人管理系统是如何工作的。

范　围	抽　象	时　间	功　能
部分	业务云	当前	业务
组织	数据库云	未来	应用程序
行业	地面		

2．向一位新员工解释制造业涉及的关键概念。

范　围	抽　象	时　间	功　能
部分	业务云	当前	业务
组织	数据库云	未来	应用程序

范　　围	抽　　象	时　　间	功　　能
行业	地面		

3．获取一份关于新的销售数据集市的详细需求（数据集市是为了满足一些特定用户需求而设计的一种数据仓库）。

范　　围	抽　　象	时　　间	功　　能
部门	业务云	当前	业务
组织	数据库云	未来	应用程序
行业	地面		

关键点

√ 照相机上有4种设置，变焦、对焦、定时器、滤镜，它们都可以被直接转换到数据模型上。变焦可以转换为数据模型的范围。对焦可以转换为数据模型的抽象。定时器转换为时间设置，用来决定数据模型获取当前的视图，还是未来的视图。过滤器转换为功能设置，用来决定数据模型获取的是业务视角，还是应用程序视角。

√ 用适当的模型设置匹配建立模型的目标，可以提升数据模型以及它所支撑的应用项目的质量。

√ 不要忘记关于图像格式的可选项！人们更喜欢去看一份校样（概念数据模型）、底片（逻辑数据模型），还是图片（物理数据模型）？

第2部分　数据模型要素

第2部分将解释数据模型中所使用的符号及文本。第4章解释实体，第5章则关于属性，第6章讨论关系，第7章说明键。当完成了本部分的学习，你将可以读懂任意规模、复杂度的数据模型。

第4章介绍了实体（entity）的定义并讨论了不同种类的实体，实体实例也将于本章介绍。同时，对实体上存在的3种层次——概念、逻辑、物理也做了相应的说明。进一步地还介绍了与弱实体（weak entity）相关的概念。

第5章介绍了属性的定义并讨论了域的概念，而且还给出了3种不同域类型的实例。

第6章介绍了规则和关系的定义，数据规则有别于行为规则。另外，基数和标签也将会被阐述。由此使得能像阅读小说那样轻松地读

懂任何数据模型。递归关系（recursive relationships）、子类型（subtyping）等关系类型也将被讨论。

第7章介绍了键的定义，并对候选键、主键、备用键等术语加以区分，而且还将介绍代理键、外键的定义，并对它们的重要性加以解析。

第4章　实体

有趣的概念

谁、什么、何时、何地、为何及如何

实体比比皆是。

当我在教室中来回踱步，想看看是否有学生会有疑问时，我注意到坐在最后一排的一名同学已经完成了练习，我走到她的座位旁，只看见她在纸上画了几个矩形框，其中有一个大点的矩形框里面写着“生产”，我询问她如何理解所定义的“生产”，她回答说：“生产是一个将原材料加工成最终产品的过程，所有的生产步骤都被包含在这个矩形框中”。

事实上，数据模型中的矩形，即实体，不是被设计用来表示或包含处理的。相反，实体是用来表示在处理中所使用到的一些概念。那名同学所设计的模型里的“生产”实体，事实上可以被最终转化成其他的几个实体，包括“原材料”“最终货物”“机器”“生产计划”等。

本章定义了实体的概念，并讨论了实体的不同种类（谁、什么、何时、何地、为何及如何），同时，对实体的3个层次—概念、逻辑、物理加以解释，进一步地，还介绍了与弱实体相关的概念。

4.1 实体的说明

一个实体表示的是对于业务非常重要或值得获取的事物及与之相关的信息集合。每个实体都由一个名词或名词词组定义，并符合六大种类之一：谁、什么、何时、何地、为何及如何。表4.1为实体种类的定义及相应的实例。

表4.1 实体信息

种类	定　义	实　例
谁	对企业有益的人或组织，即“业务中，谁是重要的？”通常人或组织与某一角色关联，如“顾客”或“供应商”	Employee、Patient、Player、Suspect、Customer、Vendor、Student、Passenger、 Competitor、Author
什么	对企业有益的产品或服务，通常可以理解为：组织会把什么保留在它的业务内，即对业务而言重要的东西是什么	Product、Service、Raw Material、Finished Good、 Course、Song、Photograph、 Title
何时	对企业有益的日程或时间间隔，即业务何时运作	Time、Date、Month、Quarter、Year、Semester、Fiscal Period、Minute
何地	对企业有益的位置，位置可以是一个实际的地点，也可以是一个电子化的虚拟场所，即业务在哪开展	Mailing Address、Distribution Point、Website URL、IP Address

种类	定　义	实　例
为何	对企业有益的事件或交易，这些事件保证业务的运转，即业务运转的原因	Order、Return、Complaint、Withdrawal、Deposit、Compliment、Inquiry、Trade、Claim
如何	对企业有益的事件的文档，文档用来记录事件，如“采购订单”里记录了一次订购事件，即在业务中事件如何被跟踪	Invoice、Contract、Agreement、Purchase Order、Speeding Ticket、Packing Slip、Trade Confirmation

实体实例是一个具体实体的呈现或者说是实体的值。试想将一个电子表格当作一个实体，其中列标题代表实体应该记录的一些信息，每个电子表格行包含的实际值则为一个实体实例。例如，实体“顾客”可以被一些如Bob、Joe、Jane等具体的姓名实例化，实体“账户”则可能有诸如Bob的支票账户、Bob的储蓄账户、Joe的经纪人账户等实例。

4.2 实体类型

数据模型之美在于你可以根据不同的受众把相同的信息以不同的细节水平呈现出来。上一章介绍了3种细节水平：概念、逻辑、物理。实体是所有3个细节水平的组成部分。

实体可以在概念、逻辑和物理3种层次上被描述。概念意味着高层次的业务流程的解决方案或应用程序频繁定义的范围和重要术语。逻辑意味着业务流程的详细解决方案或应用程序。物理意味着应用程序详细的技术解决方案。

那些基本、关键的业务信息，才能与实体的概念层相关，而什么是基本且关键的信息，这很大程度上取决于所关注的范围。在一个普遍的范围内，有一些最常见的共识概念，例如，“顾客”“产品”和“员工”。如果将范围缩小一点，一个给定的行业可能会产生一些特定的概念，对于广告行业，“宣传”可以是一个有效的概念，但对于其他行业则不尽然。

在逻辑层上描述的实体，使用了比概念层更多的细节来描述业务。通常，一个概念实体可以被表示成多个逻辑数据模型实体，逻辑实体中包含的属性（attributes）将在第5章讨论。

在物理层上，实体对应于某种特定技术的对象。例如，关系型数据库管理系统RDBMS中的数据库表，又如NoSQL数据库MongoDB中的集合（collection）。物理层与逻辑层非常相似，但是往往需要一些技术在数据库执行性能及数据存储上找到相应的解决方案。物理实体

还包含一些与特定数据库相关的信息，例如，属性的格式或长度（作者的姓氏，长度50个字符），或者属性是否需要被赋值（作者税号不为空，故需要赋值，作者生日可为空，故可以不赋值）。

在关系型数据库（RDBMS）中，物理实体对应于数据库表或视图。而在NoSQL数据库中，物理实体的转换取决于底层技术，例如，在一个基于文档的数据库MongoDB中，实体对应于集合（collection）。而通用术语结构（structure）指的是底层数据库组件，与具体的RDBMS或NoSQL数据库解决方案无关。

图4.1所示为几个与冰淇淋店有关的实体，每个实体用包含实体名的矩形框表示。

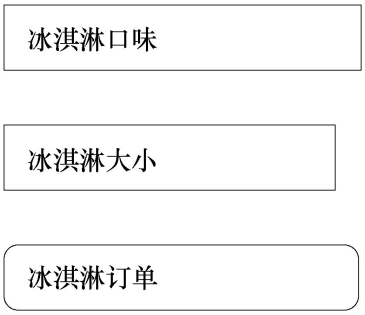

图4.1　实体的表示

需要注意的是有两种类型的矩形框，例如，冰淇淋口味、冰淇淋大小那样的直角矩形框，还有如冰淇淋订单那样的圆角矩形框。这里并不打算用过时的建模术语来区分两种矩形框，只需明确对于大多数建模工具来说，直角框表示强实体，圆角框表示弱实体。

强实体可以独立存在，用来表示相对独立的人、事或地点。例如，为了检索某位特定顾客的信息，可以在数据库中使用顾客号进行

查找。“这是Bob，顾客号为123”。巧克力风味的冰淇淋可以用C进行检索，冰淇淋大小为两匙的信息可以用数字2进行检索。

弱实体至少依赖于一个其他的实体，这意味着如果不引用其他实体的实例，就无法检索弱实体的实例，例如，冰淇淋订单可以由冰淇淋口味或冰淇淋大小，再结合冰淇淋订单中的某些内容（如序号）进行检索。

数据模型是一种交流工具。理解强实体、弱实体间的差别将有助于我们理解实体间的关系和依赖。例如，在阅读数据模型时发现冰淇淋订单是依赖于冰淇淋口味的弱实体，于是在软件开发过程中就应该确保冰淇淋口味信息先于订单提交被添加，即提交一份巧克力冰淇淋订单之前，作为冰淇淋口味的“巧克力”需要在软件系统中可用。

4.3 练习4：定义概念

列举3个你所在机构的概念。机构中对这3个概念是否有唯一共识的定义？如果不是，为什么？你是否可以为每一条给出一个单独的定义？

> **关键点**
>
> √ 一个实体表示的是对于业务非常重要或值得获取的事物及与之相关的信息集合。实体应该符合六大种类之一：谁、什么、何时、何地、为何及如何。
>
> √ 实体由名词或名词词组定义。
>
> √ 实体实例是一个具体实体的呈现或者说是实体的值。
>
> √ 实体可以存在于概念、逻辑、物理等3种细节水平上。
>
> √ 实体可分为强实体和弱实体。

第5章　属性

电子表格由各列构成，

属性类似于列，

模型无处不在。

本章介绍属性的概念及属性可存在的3个不同层次—概念、逻辑、物理。域及不同类型的域也将被讨论。

5.1 属性的解释

属性是一则相对独立的信息，其值用以识别、描述、评估实体实例。例如，属性“索赔号”可以识别每个索赔，属性“学生的姓氏”用来描述学生。属性“销售总额”用来评估交易中获取的财政收入。

以电子表格为例，电子表格中的列标题就是属性。每个列标题下方一个个单元格用来存储相应属性的值。我们可以将电子表格中的列标题、表单中的域、报表中的标签都理解为属性。“冰淇淋风味名”“冰淇淋大小代码”是关于冰淇淋店的属性，而“公司名”“电话号码”是关于名片的属性。

5.2 属性类型

与实体类似，属性也可以在概念、逻辑、物理等3个层次上加以描述。概念级属性必须是对业务起着基本且又关键影响的概念。一般情况下，属性不被当作概念，但这取决于业务需求，允许例外。以前，我曾为一家通信公司提供数据建模服务，在其他应用中电话号码通常被视为属性，但它对于这家通信公司的业务却非常重要，所以电话号码被表示成了概念数据模型中的概念。

逻辑模型中的属性则描述的是业务特征。每个属性对于业务解决方案都有不同程度的贡献，并且与任何软、硬件技术无关。例如，“冰淇淋口味名”就是一则逻辑级属性，因为它对业务解决方案有重要意义，而且并不取决于到底存储在纸质文件中，还是存储在高速数据库中。与物理数据模型对应的属性可以被理解为一个物理“容器”，用来存储数据，属性“冰淇淋口味名”在RDBMS中可以被表示为ICE*CRM表中的ICE_CRM_FLVR* NAM列，或者在MongoDB数据库中被表示为IceCream集合中的字段IceCreamFlavorName。

需要注意的是本书中为了保持文字上的一致性，我们使用的是“属性”（attribute）。但在实际工作中，我则建议使用那些更容易让用户接纳的术语。例如，有的业务分析师可能更倾向于使用特征（property）或标签（label），而有的数据库管理员或许更习惯使用列（column）或字段（field）。

5.3 域的解释

域是某一属性所有可能取值的集合。域中往往还包含一组验证标准，使得域可以被多个属性使用。例如，“日期”域中包括所有的合法日期，它可以被应用于以下这些属性。

- 雇员入职日期
- 订单输入日期
- 索赔提交日期
- 课程开始日期

如果属性与域相关联，那么该属性的取值绝对不允许超出该域，域中的值可以由一组特定的数据列表指定，也允许由一组规则指定。例如，“员工性别”可以由取值为“男”和“女”的域限定。“员工入职日期”可以由一组规则限定，如取规则为“合法日期”，则其可能取值如下。

- February 15th,2005
- 25 January 1910
- 20150410
- March 10th,2050

由于员工入职日期应该被设定为一个有效的日期，故February 30th被排除。在此基础上，还可以用一组附加规则来限定其域。例如，限定员工入职日期的域为早于今天，这样March 10th,2050被排除，又如果限定其格式为YYYYMMDD（年、月、日串联日期格

式），除了20150410之外其他的都应被排除。还可以使用精简的数据集合来限定员工入职日期的域，即规定该日期必须符合星期一、星期二、星期三、星期四、星期五中的一个（典型的工作日）。

在名片实例中，“联系人姓名”可能包含数千种，甚至数百万种取值，如图1.2给出的4张名片，其姓名为：

- Steve Hoberman
- Steve
- Jenn
- Bill Smith
- Jon Smith

姓名域应该需要稍作精简，有必要明确此域的域值是否必须由姓和名两部分构成，如Steve Hoberman，还是可以仅包含名，如Steve。该域可以包含公司名吗，如IBM？这个域是否允许出现数字，而不仅仅是字母，如来自电影星际大战的名字R2D2？这个域是否可以出现一些特殊的字符，如O(+>?O(+>，该字符串是音乐王子在1993年把他的名字变成这种不能发音的“爱的符号”。

以下为3种基本的域类型。

① 格式域将数据指定为数据库中的标准类型，如整型（Integer）、字符型（Character（30））、日期（Date）等都是格式域。

② 列表域类似于一个下拉列表，它由一个可选的有限值的集合组成，列表域是格式域的精简，如“订单状态代码”的格式域可以被置为Character(10)，在此基础上该域可以由一个（Open、Shipped、Closed、Returned）列表域进一步精简。

③ 范围域的设置要求取值介于最小值与最大值之间，例如，“订单交付日期”必须为从今天到未来3个月中的某天。与列表域类似，范围域也是格式域的精简。

基于以下几个原因，域是非常有用的。

① 插入数据前，通过域的检查来提高数据质量。这是域存在的主要原因，通过限定属性的可能取值来降低脏数据进入数据库的可能性。例如，每一个表示金额的属性被设置为“数量域”，该域要求数字的长度上限为15且包括小数点后的两位，显然这是表示货币数额很好的一种方法，“销售总额”若被设置为“数量域”，则不允许如R2D2这样的值被添加。

② 数据模型的交流性更强。当我们在数据模型上设置了域，就意味着数据模型的一个属性必须具备一个特定域的特征，这样数据模型就变成更容易被理解的交流工具。例如，我们可以让“销售总额”“净销售额”“标价销售额”3个属性都可以共享一个“数量域”，进而共享域的特征，它们的取值都被限定为“货币”。

③ 使得新建模型、维护现有模型变得更有效率。当一位模型构建师开始一项新工程时，可以使用一组标准域来节省时间，而无需重新

创建。例如，所有与数量有关的属性，都可以同时与数量域关联，这样可以极大节省分析、设计时间。

5.4 练习5：设置域

为下列3个属性设置适当的域?

- 电子邮件地址
- 销售总额
- 国家代码

> **关键点**
>
> √ 对业务而言，属性是非常重要性的特征，其值用以识别、描述、评估实体实例。
>
> √ 域中往往包含一组验证标准，使得域可以被多个属性应用。
>
> √ 域的不同类型包括：格式域、列表域、范围域。

第6章　关系

规则无处不在，

关系讲述着故事，

并把一个个情节联系起来。

本章介绍了规则和关系的定义，以及关系存在的3个层次，概念、逻辑、物理。数据规则有别于行为规则。基数及标签也将在本章阐述。学习完本章你可以像读书那样读懂任何数据模型。递归关系（recursive relationships）和子类型（subtyping）等关系类型也将被讨论。

6.1 关系的解释

通常我们对规则的理解是在特定情形下如何行为的规定和指示。以下列举了你应该非常熟悉的关于规则的例子。

- 在你外出玩耍之前，房间必须被整理干净。
- 如果击球手3次挥棒不中，则三振出局，轮到下一位击球手回合。
- 限速每小时55英里（1英里≈1.61千米）。

数据模型中的规则即为关系，关系被表示成一条连接两个实体的线段，用来说明实体间的规则或导航路径。如果两个实体分别为“Employee”（员工）和“Department”（部门），则关系可以描述的规则有“每位员工必须服务于一个部门”“一个部门可以拥有一位或多位员工”。

6.2 关系的类型

规则可以是数据规则，也可以是行为规则。数据规则指示数据间如何关联，行为规则指示当属性包含有某特定值时，需要采取什么操作，下面首先介绍数据规则。

存在两种类型的数据规则，结构完整型（structural integrity，SI）和参照完整型（referential integrity，RI）。结构规则（又被称为基数规则）定义了参与某个关系的实体实例的数量，例如：

- 每种产品可以出现在一个或多个订单行上。
- 每个订单行上有且仅有一则产品。
- 每位学生必须有唯一的学号。